调优案例分析与实战

2022年1月10日 16:18

大内存硬件上的程序部署策略.

① 问题本质.

过大堆内存进行回收时带来的长时间的停顿.

② 部署方式.

① 一个JVM实例管理大量Java堆内存.

可能面临的挑战:

① 回收大块内存导致STW过长.

Solution: G1 → 增量回收较好应用.

ZGC、Shenandoah成熟可彻底解决.

② 必须 64bit JVM支持.

由于压缩指针、Cache Line容量等因素 64bit比32bit性能↓.

③ 要保证应用稳定.

堆内存太大,若OOM,几乎无法产生堆转储快照.

④ 64bit比32bit内存消耗大.

因为指针膨胀,数据类型对齐补白. → 不对齐读一个数据可能要读两次

↘ 可以开启指针压缩.

② 多个JVM实例,建立逻辑集群.

① 节点竞争全局资源.(例:I/O).

② 资源池利用率↓.

③ 32bit内存大小受限.

④ 大量使用本地缓存的应用 → 内存浪费.

③ 控制FullGC频率.

关键: 老年代相对稳定

↘ 取决于:应用中大部分对象"朝生夕死".

B/S应用中. 多数对象生命周期是请求级、页面级的. → 请求峰值.

→ 以此估算年轻代大小.

会话级、会局级较少.

实战调优

实战优化.

① 升级JDK及兼容问题.

② 编译时间和类加载时间优化. 取消字节码校验.

③ 调整内存设置控制GC频率. Jstat -gcutil vmid.
-gccause vmid.

④ 选择垃圾收集器降低延迟.